



AF IFW  
2131

ATTORNEY DOCKET NO: E0295.70188US00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Paul R. Carpentier et al.  
Serial No.: 09/391,360  
Filed: September 10, 2002  
For: SYSTEM AND METHOD FOR SECURE STORAGE,  
TRANSFER AND RETRIEVAL OF CONTENT  
ADDRESSABLE INFORMATION

Examiner: Leynna A. Ha  
Art Unit: 2131

Confirmation No: 8493

---

**CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)**

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the 21 day of July, 2004.

  
Signature

---

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Transmitted herewith are the following documents:

- ☒ Request For Reconsideration
- ☒ Article entitled "Applied Cryptography" by Bruce Schneier
- ☒ Return Receipt Postcard

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned at (617) 720-3500, Boston, Massachusetts.

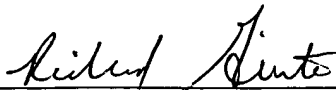
Serial No.: 09/391,360

-2-

Art Unit: 2131

A check is not enclosed. If a fee is required, the Commissioner is hereby authorized to charge Deposit Account No. 23/2825. A duplicate of this sheet is enclosed.

Respectfully submitted,  
*Paul R. Carpentier et al., Applicant*

By:   
Richard F. Giunta, Reg. No.: 36,149  
Wolf, Greenfield & Sacks, P.C.  
600 Atlantic Avenue  
Boston, Massachusetts 02210-2211  
Telephone: (617)720-3500

Docket No. E0295.70188US00

Date: July 21, 2004

**xNDDx**



ATTORNEY DOCKET NO: E0295.70188US00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE


Applicant: Paul R. Carpentier et al.  
Serial No.: 09/391,360 Confirmation No.: 8493  
Filed: September 10, 2002  
For: SYSTEM AND METHOD FOR SECURE STORAGE,  
TRANSFER AND RETRIEVAL OF CONTENT  
ADDRESSABLE INFORMATION

Examiner: Leynna A. Ha  
Art Unit: 2131 Confirmation No: 8493

---

**CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)**

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the 21 day of July, 2004.

  
Signature

---

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

**REQUEST FOR RECONSIDERATION**

In response to the Office Action of May 21, 2004, Applicants respectfully request reconsideration. To further the prosecution of this application, each of the rejections set forth in the Office Action is addressed below. The application as presented is believed to be in condition for allowance.

Initially, Applicants' representatives thank Examiner Ha for the courtesies extended in granting and conducting a telephone interview on July 15, 2004. The substance of the interview is summarized herein.

The Office Action rejected claims 1-5, 7-10, 13, 14, 18, 20-33 under 35 U.S.C. §102(e) as purportedly being unpatentable over Saito (6,076,077) and claims 6, 11, 12, 15-17 and 19 under 35 U.S.C. §103(a) as purportedly obvious over Saito in view of various other references. Applicants respectfully traverse each of these rejections.

### **Rejections Under 35 U.S.C. §102**

#### **Claim 1**

Claim 1 is directed to a method comprising: generating a first unique identifier for a binary asset, said first unique identifier being computed from at least a portion of the contents of said binary asset and uniquely identifying said binary asset; and encrypting said binary asset using said first unique identifier as a key, said encrypting resulting in an encrypted version of said binary asset.

#### **A. No Explicit Disclosure In Saito of Encrypting An Asset Using A Key Computed From At Least A Portion Of The Contents Of The Asset**

During the telephone interview, Applicants pointed out that Saito does not disclose or suggest encrypting a binary asset using a unique identifier computed from at least a portion of the contents of the binary asset as a key. In this respect, Applicants' representatives explained that they had reviewed the sections of Saito cited by the Office Action as purportedly disclosing this limitation, in addition to the rest of Saito, and saw no disclosure whatsoever as to how the encryption keys are generated, let alone a disclosure of generating an encryption key based on the content of the asset being encrypted. During the telephone interview, the Examiner looked again at Saito for any disclosure of the encryption key being generated based upon the content of the asset. The Examiner could not point to any such disclosure, but understandably felt rushed and agreed to look at the issue again in response to this request for reconsideration. As discussed during the telephone interview, if the rejection is to be maintained, it is respectfully requested that the Examiner provide a citation to the specific portion(s) of Saito that is believed to disclose encrypting a binary asset using a unique identifier computed from at least a portion of the contents of the binary asset as a key.

**B. The Use Of An Encryption Key Computed From At Least A Portion Of The Contents Of The Asset Is Not Inherent**

---

During the telephone interview, the Applicants' representatives suggested that they were confident that upon a complete review of Saito, the Examiner would not find any disclosure about encrypting an asset using an encryption key generated based at least in part on the content of the asset. Thus, it was agreed that the discussion should continue assuming that that were the case, to address whether the Examiner would agree that the claims would then distinguish over Saito. Specifically, Applicants' representatives indicated that it would be helpful to talk through the issue, particularly in view of the "Response to Amendment" section of the Final Office Action, which indicated that certain features were inherently disclosed in Saito or potentially were obvious.

Initially, the Examiner clarified that the position expressed in the Office Action was that, to the extent not explicitly disclosed in Saito, it was believed that encrypting an asset using an identifier based at least in part on the contents of the asset was inherent in Saito, and that despite the reference in the "Response to Amendment" section to it purportedly being "obvious that the key is computed from the content of the data" the rejection was based on Saito (either explicitly or inherently) alone, and that is why the rejection was made under §102 rather than under §103.

During the telephone interview, the Examiner defended the assertion, in the Response to Amendment section of the Office Action (paragraph 9, page 13), that it is purportedly inherent that an encryption key identifies a record in a data file because a key is inherently unique. In support of this assertion, the Examiner cited the Microsoft Computer Dictionary, 5<sup>th</sup> edition, page 300, which defines the term "key" (significantly, for usage in the *database management context* as opposed to encryption) as, "[i]n database management, an identifier for a record or group of records in a datafile." Applicants respectfully disagree with any assertion that an encryption key inherently uniquely identifies a data file.

The definition of a key cited by the Examiner pertains to database management. A key in the context of a database management system is very different from an encryption key. In a database management system, a key is a field in a database table on which the data in the table is sorted. For example, if a database table had three fields, such as "first name," "last name," and

“telephone number,” and the key field was “last name,” the database table would be sorted based on the “last name” field. The use of the term key in the context of a database is unrelated to encryption and is a very different concept from an encryption key.

Indeed, the Microsoft Computer Dictionary provides an alternate definition of a key that relates specifically to the field of encryption: “[i]n encryption and digital signatures, a string of bits used for encrypting and decrypting information to be transmitted. Encryption commonly relies on two different types of keys, a public key known to more than one person (say, both the sender and the receiver) and a private key known only to one person (typically, the sender).” If the Examiner continues to rely on the Microsoft Computer Dictionary, it is respectfully requested that the Examiner apply the definition of “key” that pertains to the field of encryption. Notably, the definition that pertains to encryption does not suggest that an encryption key is inherently unique or inherently identifies a data file.

In view of the foregoing, any assertion that it is inherent that an encryption key uniquely identifies a data file is respectfully traversed. MPEP §2112 states that, “[t]o establish inherency, the extrinsic evidence ‘must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.’” MPEP, 8<sup>th</sup> Edition, Rev. 1, Feb. 2003, page 2100-52. Indeed, because a single key is commonly used to encrypt more than one data file, it is not inherent that a key identifies a data file nor is it inherent that a key is unique. An example of a situation in which a single key may be used to encrypt multiple data files is public key encryption, which is disclosed in Saito. In public key encryption, a person receiving encrypted data files publishes a public key. The public key has a corresponding private key which is kept secret by the person receiving data files. Multiple different senders may use the public key to encrypt data files to be sent to the person receiving the data files. These encrypted data files may only be decrypted by the receiver’s secret private key. Thus, any one who wishes to send data to the receiver may encrypt such data using the receiver’s public key. Only the person with the secret private key (i.e., the receiver) can decrypt these encrypted data files. As a result, the public key is used by multiple different people to encrypt many different files and does not identify any particular data file.

As mentioned above, to establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference. Here, it is simply not necessary that the encryption keys of Saito be generated based on the content of the data that they are used to encrypt, as there are many other methods for generating keys.

For example, enclosed are a few pages from the textbook Schneier, Bruce, "Applied Cryptography", 2<sup>nd</sup> Edition, page 173, which has a section on desirable key selection. Not only does this text not suggest that encryption keys be based on any content, but it teaches away from such a key generation technique by suggesting that keys should be random:

[g]ood keys are random-bit strings generated by some automatic process. If the key is 64 bits long, every possible 64-bit key must be equally likely. Generate the key bits from either a reliable random source...or a cryptographically secure pseudo-random bit generator...If these automatic processes are unavailable, flip a coin or roll a die. Page 173

As should be appreciated from the foregoing, there are known techniques for generating encryption keys that do not generate a key based upon the content of any data to be encrypted (e.g., the use of a random number). Thus, it does not necessarily follow from the disclosure in Saito that an encryption key must necessarily be generated based upon any content to be encrypted. Therefore, it is respectfully asserted that to the extent the rejection of claim 1 is based upon Saito purportedly inherently teaching an encryption key based at least in part on the content of an asset to be encrypted, the rejection is improper.

As should be clear from the discussion above, Saito does not disclose or suggest, either explicitly or inherently "generating a first unique identifier for a binary asset, said first unique identifier being computed from at least a portion of the contents of said binary asset and uniquely identifying said binary asset" and "encrypting said binary asset using said first unique identifier as a key, said encrypting resulting in an encrypted version of said binary asset," as recited in claim 1. Accordingly, it is respectfully requested that the rejection of claim 1 under 35 U.S.C. §102(e) be withdrawn.

Claims 2 and 3 depend from claim 1 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 2 and 3 be withdrawn.

**Claim 4**

Claim 4 is directed to a method comprising: generating a first file identifier for a file, said first file identifier being computed from at least a portion of said file and uniquely identifying said file; encrypting said file using said first file identifier as a key, said encrypting producing an encrypted file; generating a second file identifier for said encrypted file, said second file identifier being computed from at least a portion of said encrypted file and uniquely identifying said encrypted file; and providing said first file identifier and said second file identifier for the retrieval of said file, whereby said second file identifier may be used to locate said encrypted file, and said first file identifier may be used to decrypt said encrypted file to produce said file.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, “generating a first file identifier for a file, said first file identifier being computed from at least a portion of said file and uniquely identifying said file” and “encrypting said file using said first file identifier as a key.” As discussed above, the keys in Saito are not computed from a portion of the content which they encrypt and do not uniquely identify a file. Accordingly, it is respectfully requested that the rejection of claim 4 under 35 U.S.C. §102(e) be withdrawn.

Claims 5-7 depend from claim 4 and are patentable for at least the same reasons. Accordingly it is respectfully requested that the rejections of claims 5-7 be withdrawn.

**Claim 8**

Claim 8 is directed to a method of uniquely and securely identifying a group of binary assets, each binary asset representing digital information. The method comprises: computing an intrinsic unique identifier (IUI) for each of said binary assets; encrypting each of said binary assets using the IUI of each asset as its key to produce an encrypted version of each of said binary assets; computing an IUI of each of said encrypted versions; creating a file that includes said IUIs of said binary assets and said IUIs of said encrypted versions; computing a key IUI for said file; encrypting said file using said key IUI to produce an encrypted file; and computing a master IUI for said encrypted file, whereby said key IUI and said master IUI uniquely represent said binary assets and may be used to locate said assets.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, “computing an intrinsic unique identifier (IUI) for each of said binary assets” and “encrypting



each of said binary assets using the IUI of each asset as its key to produce an encrypted version of each of said binary assets,” as recited in claim 8. Accordingly, it is respectfully requested that the rejection of claim 8 under 35 U.S.C. §102(e) be withdrawn.

Claims 9-12 depend from claim 8 and are patentable for at least the same reasons. Accordingly it is respectfully requested that the rejections of claims 9-12 be withdrawn.

### **Claim 13**

Claim 13 is directed to a descriptor file data structure that reliably identifies a plurality of files. The data structure comprises: a file name for each of said files; meta data for each file indicating attributes of each file; a first intrinsic unique identifier (IUI) for each of said files, each IUI being calculated from the contents of its corresponding file and uniquely identifying its corresponding file; and a second IUI associated with each of said files, each second IUI being calculated from an encrypted version of its associated file, each file being encrypted using its associated first IUI as a key, wherein said second IUIs may be used to locate said encrypted versions of said files, and said first IUIs may be used to decrypt said encrypted versions to obtain the non-encrypted versions of said files.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, a data structure that identifies a plurality of files comprising “a first intrinsic unique identifier (IUI) for each of said files, each IUI being calculated from the contents of its corresponding file and uniquely identifying its corresponding file” and “a second IUI associated with each of said files, each second IUI being calculated from an encrypted version of its associated file, each file being encrypted using its associated first IUI as a key,” as recited in claim 13. Accordingly, it is respectfully requested that the rejection of claim 13 under 35 U.S.C. §102(e) be withdrawn.

Claim 14 depends from claim 13 and is patentable for at least the same reasons. Accordingly it is respectfully requested that the rejection of claim 14 be withdrawn.

### **Claim 18**

Claim 18 is directed to a method of reliably retrieving a secure file. The method comprises: receiving an intrinsic unique identifier for an encrypted version of said file; retrieving said encrypted version of said file using said IUI of said encrypted versions; receiving an IUI for

the non-encrypted version of said file; and decrypting said encrypted version of said file using said IUI of said non-encrypted version as a key to obtain the non-encrypted version of said file, whereby said IUI of said encrypted version and said IUI of said non-encrypted version provide access to the contents of said file.

Saito fails to disclose or suggest, *inter alia*, “decrypting said encrypted version of said file using said IUI of said non-encrypted version as a key to obtain the non-encrypted version of said file” as recited in claim 18. Nowhere does Saito disclose or suggest decrypting an encrypted version of a file using an IUI of the non-encrypted version as a key. Accordingly, it is respectfully requested that the rejection of claim 18 under 35 U.S.C. §102(e) be withdrawn.

Claims 19 and 20 depend from claim 18 and are patentable for at least the same reasons. Accordingly, it is respectfully requested the rejections of claims 19 and 20 be withdrawn.

### **Claim 21**

Claim 21 is directed to a method of obtaining a data file that has been securely stored. The method comprises: receiving a master identifier that uniquely identifies an encrypted file; retrieving said encrypted file using said master identifier; receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted file; decrypting said encrypted file using said key identifier to obtain said non-encrypted version, said non-encrypted version including a data file identifier that uniquely identifies a data file and an encrypted version of said data file; retrieving said encrypted version of said data file using said encrypted identifier; and decrypting said encrypted data file using said data file identifier as a decryption key, whereby said non-encrypted version of said data file is obtained.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, “receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted file” and “decrypting said encrypted file using said key identifier to obtain said non-encrypted version,” as recited in claim 21. Accordingly, it is respectfully requested that the rejection of claim 21 under 35 U.S.C. §102(e) be withdrawn.

Claims 22-25 depend from claim 21 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 22-25 be withdrawn.

**Claim 26**

Claim 26 is directed to a method of obtaining a data file that has been securely stored. The method comprises: receiving a user identifier that uniquely identifies a non-encrypted first file, said non-encrypted first file including a unique identifier identifying an encrypted version of said data file and a master identifier that uniquely identifies an encrypted version of a descriptor file; retrieving said non-encrypted first file using said user identifier; retrieving said encrypted descriptor file using said master identifier; retrieving said encrypted data file using said unique identifier for said encrypted version of said data file; receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted descriptor file; decrypting said encrypted descriptor file using said key identifier to obtain said non-encrypted version of said descriptor file, said non-encrypted version including a data file identifier that uniquely identifies said data file; and decrypting said encrypted data file using said data file identifier as a decryption key, whereby said non-encrypted version of said data file is obtained.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, “receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted descriptor file” and “decrypting said encrypted descriptor file using said key identifier to obtain said non-encrypted version of said descriptor file,” as recited in claim 26. Accordingly, it is respectfully requested that the rejection of claim 26 under 35 U.S.C. §102(e) be withdrawn.

Claims 27-30 depend from claim 26 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 27-30 be withdrawn.

**Rejections Under 35 U.S.C. §103**

The Office Action rejected claims 6, 11, and 19 under 35 U.S.C. §103(a) as purportedly being obvious over Saito in view of Berkowitz (5,832,479) and claims 12 and 15-17 as being purportedly obvious over Saito in view of Microsoft Computer Dictionary, 5<sup>th</sup> Edition. Applicants respectfully traverse each of these rejections.

Claims 6, 11, 12, and 19 are dependent claims, and each is patentable for at least the same reasons as the claim from which it depends. Claims 15-17 are discussed below.

**Claim 15**

Claim 15 is directed to a method of uniquely and securely identifying a group of files. The method comprises: creating a key file that includes a plurality of cryptographic keys, each key being associated with one of said group of files; computing a unique identifier for said key file, said key file identifier being calculated from a portion of the contents of said key file; encrypting said key file using said key file identifier to produce an encrypted key file; computing a unique identifier for said encrypted key file, said encrypted key file identifier be calculated from a portion of the contents of said encrypted key file; creating a flattened file that includes said encrypted key file identifier and unique identifiers for encrypted version of said files, each unique identifier of one of said encrypted files being calculated from the contents of its associated encrypted file, each encrypted file having been encrypted using its associated key to encrypted the plaintext version of the file; and computing a user unique identifier for said flattened file, said user unique identifier be calculated from a portion of the contents of said flattened file, whereby a user provided with said user unique identifier may retrieve said flattened file and said encrypted versions of said files, and when provided with said key file identifier said user may decrypt said encrypted files.

The Office Action asserts that Saito discloses all the limitations of claim 15, except that Saito fails to explicitly teach a flattened file. The Office Action further asserts that the Microsoft Computer Dictionary discloses a flattened file and that it would have been obvious to use a flattened file in the system of Saito “because it significantly reduces file sized (sic) and can be saved in a wider range of formats.” (*see* Pages 12-13 of Office Action). Applicants respectfully disagree with these assertions.

As should be clear from the discussion above, Saito fails to disclose or suggest, *inter alia*, “computing a unique identifier for said key file, said key file identifier being calculated from a portion of the contents of said key file” and “encrypting said key file using said key file identifier to produce an encrypted key file,” as recited in claim 15. This deficiency is not remedied by the Microsoft Computer Dictionary. Accordingly, it is respectfully requested that the rejection of claim 15 under 35 U.S.C. §103 be withdrawn.

Claims 16 and 17 depend from claim 15 and are patentable for at least the same reasons. Accordingly, it is respectfully requested that the rejection of claims 16 and 17 be withdrawn.

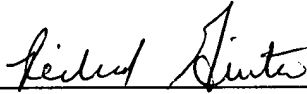
**CONCLUSION**

In view of the foregoing remarks, this application should now be in condition for allowance. A notice to this effect is respectfully requested. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below to discuss any outstanding issues relating to the allowability of the application.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully submitted,

*Paul R. Carpentier et al., Applicant*

By:   
Richard F. Giunta, Reg. No. 36,149  
Wolf, Greenfield & Sacks, P.C.  
600 Atlantic Avenue  
Boston, Massachusetts 02210-2211  
Telephone: (617) 720-3500

Docket No. E0295.70188US00

Date: July 21, 2004

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

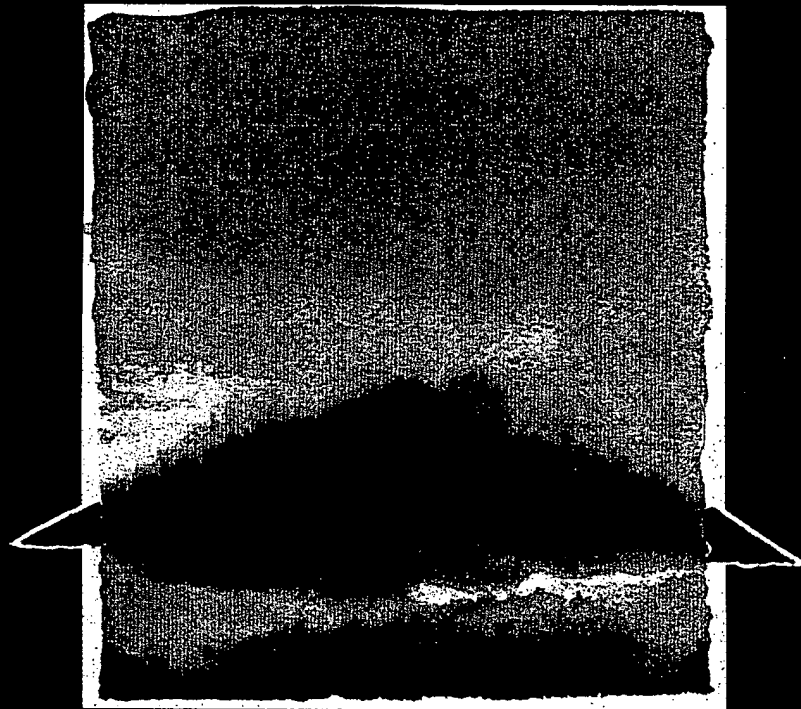
**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

*...the best introduction  
to cryptography I've  
ever seen. It's the book  
the National Security  
Agency wanted, rather  
than to be published.*

*—Wired Magazine*

**SECOND  
EDITION**

# **APPLIED CRYPTOGRAPHY**



**Protocols, Algorithms,  
and Source Code in C**

**BRUCE SCHNEIER**

# APPLIED CRYPTOGRAPHY, SECOND EDITION

PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C

BRUCE SCHNEIER



John Wiley & Sons, Inc.

New York • Chichester • Brisbane • Toronto • Singapore



names. The Pinyin Romanization of Chinese syllables was used, combining syllables together into one-, two-, and three-syllable words. Because no tests were done to determine whether the words actually made sense, an exhaustive search was initiated. Since there are 298 Chinese syllables in the Pinyin system, there are 158,404 two-syllable words, and slightly more than 16,000,000 three-syllable words. A similar mode of attack could as easily be used with English, using rules for building pronounceable non-sense words.

6. Word pairs. The magnitude of an exhaustive test of this nature is staggering. To simplify the test, only words of three or four characters in length from `/usr/dict/words` were used. Even so, the number of word pairs is about ten million.

A dictionary attack is much more powerful when it is used against a file of keys and not a single key. A single user may be smart enough to choose good keys. If a thousand people each choose their own key as a password to a computer system, the odds are excellent that at least one person will choose a key in the attacker's dictionary.

### Random Keys

Good keys are random-bit strings generated by some automatic process. If the key is 64 bits long, every possible 64-bit key must be equally likely. Generate the key bits from either a reliably random source (see Section 17.14) or a cryptographically secure pseudo-random-bit generator (see Chapters 16 and 17.) If these automatic processes are unavailable, flip a coin or roll a die.

This is important, but don't get too caught up in arguing about whether random noise from audio sources is more random than random noise from radioactive decay. None of these random-noise sources will be perfect, but they will probably be good enough. It is important to use a good random-number generator for key generation, but it is far more important to use good encryption algorithms and key management procedures. If you are worried about the randomness of your keys, use the key-generating technique described below.

Some encryption algorithms have weak keys: specific keys that are less secure than other keys. I advise testing for these weak keys and generating a new one if you discover one. DES has only 16 weak keys out of  $2^{56}$ , so the odds of generating one of these keys are incredibly small. It has been argued that a cryptanalyst would have no idea that a weak key is being used and therefore gains no advantage from its use. It has also been argued that not using weak keys gives a cryptanalyst more information. However, testing for the few weak keys is so easy that it seems prudent not to do so.

Generating keys for public-key cryptography systems is harder, because often the keys must have certain mathematical properties (they may have to be prime, be a quadratic residue, etc.). Techniques for generating large random prime numbers are given in Section 11.5. The important thing to remember from a key management point of view is that the random seeds for those generators must be just that:

Generating a random key isn't always possible. Sometimes you need to remember your key. (See how long it takes you to remember 25e8 56f2 e8ba c826...) To generate an easy-to-remember key, make it obscure. The ideal would be something easy to remember, but difficult to guess. Here are some suggestions:

- Word pairs separated by a punctuation character, for example, "apple\*moose" or "zorch!splat"
- Strings of letters that are an acronym of a longer phrase. For example, "Mein Luftkissenfahrzeug ist voller Aale!" generates the key "MLivA!"

### Pass Phrases

A better solution is to use an entire phrase instead of a word, and to convert the phrase into a key. These phrases are called **pass phrases**. A technique called **crunching** converts the easy-to-remember phrases into random keys. Use a one-way hash function to transform an arbitrary-length text string into a pseudo-random string.

For example, the easy-to-remember text string:

My name is Ozymandias, king of kings. Look on my works, ye mighty, and despair.

might crunch into this 64-bit key:

e6c1 4398 5ae9 0a9b

Of course, it can be difficult to type an entire phrase into a computer with the echo turned off. Clever suggestions to solve this problem would be appreciated.

If the phrase is long enough, the resulting key will be random. Exactly what "long enough" means is open to interpretation. Information theory tells us that standard English has about 1.3 bits of information per character (see Section 11.1). For a 64-bit key, a pass phrase of about 49 characters, or 10 normal English words, should be sufficient. As a rule of thumb, figure that you need five words for each 4 bytes of key. That's a conservative assumption, since it doesn't take into account case, spacing, and punctuation.

This technique can even be used to generate private keys for public-key cryptography systems: The text string could be crunching into a random seed, and that seed could be fed into a deterministic system that generates public-key/private-key pairs.

If you are choosing a pass phrase, choose something unique and easy-to-remember. Don't choose phrases from literature—the example from "Ozymandias" is a bad one. Both the complete works of Shakespeare and the dialogue from *Star Wars* are available on-line and can be used in a dictionary attack. Choose something obscure, but personal. Include punctuation and capitalization; if you can, include numbers and non-alphanumeric symbols. Poor or improper English, or even a foreign language, makes the pass phrase less susceptible to a dictionary attack. One suggestion is to use a phrase that is "shocking nonsense": something offensive enough that you are likely to remember and unlikely to write down.

everything  
keys are ra

### Key Gen

ANSI X9.17

not gener  
or pseud  
generate ke  
[X] be tripl  
key gener  
key  $R_i$ , cal

$$R_i =$$

to generate  $V_i$

$$V_i =$$

To turn  $R_i$  into  
a bit key, use it  
to generate them tog

### DoD Key Gen

The U.S. Depart  
ment 9.8) to genera  
ectors, system  
ector from the s  
externally ge  
administrator, fo

## 8.2 NONLI

Imagine that you  
topography equipr

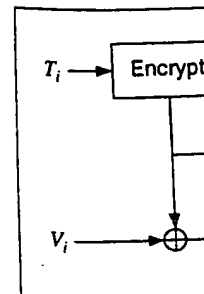


Figure 8.1 AN

Despite everything written here, obscurity is no substitute for true randomness. The best keys are random keys, difficult as they are to remember.

### X9.17 Key Generation

The ANSI X9.17 standard specifies a method of key generation (see Figure 8.1) [55]. This does not generate easy-to-remember keys; it is more suitable for generating session keys or pseudo-random numbers within a system. The cryptographic algorithm used to generate keys is triple-DES, but it could just as easily be any algorithm.

Let  $E_K(X)$  be triple-DES encryption of  $X$  with key  $K$ . This is a special key reserved for secret key generation.  $V_0$  is a secret 64-bit seed.  $T$  is a timestamp. To generate the random key  $R_i$ , calculate:

$$R_i = E_K(E_K(T_i) \oplus V_i)$$

To generate  $V_{i+1}$ , calculate:

$$V_{i+1} = E_K(E_K(T_i) \oplus R_i)$$

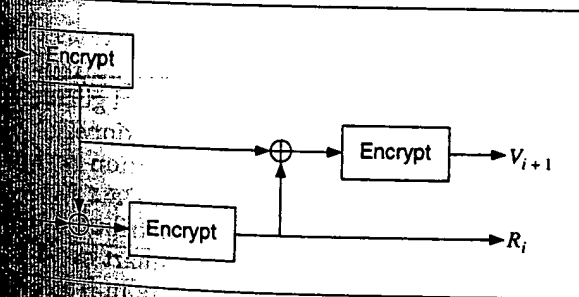
To turn  $R_i$  into a DES key, simply adjust every eighth bit for parity. If you need a 64-bit key, use it as is. If you need a 128-bit key, generate a pair of keys and concatenate them together.

### DoD Key Generation

The U.S. Department of Defense recommends using DES in OFB mode (see Section 9.8) to generate random keys [1144]. Generate a DES key from system interrupt vectors, system status registers, and system counters. Generate an initialization vector from the system clock, system ID, and date and time. For the plaintext, use an externally generated 64-bit quantity: eight characters typed in by a system administrator, for example. Use the output as your key.

## 2. NONLINEAR KEYSACES

Imagine that you are a military cryptography organization, building a piece of cryptographic equipment for your troops. You want to use a secure algorithm, but you are



ANSI X9.17 key generation.